User's Guide for the Artifact for Taxonomist: Application Detection Through Rich Monitoring Data

Emre Ates[1], Ozan Tuncer[1], Ata Turk[1], Vitus J. Leung[2], Jim Brandt[2], Manuel Egele[1], and Ayse K. Coskun[1]

[1]Dept. of Electrical and Computer Engineering, Boston University
[2]Sandia National Laboratories

July 3, 2018

1 Introduction

The artifact's format is a Jupyter Notebook. The non-interactive version of the notebook is also included as an HTML file. Our Euro-Par paper can be downloaded from `http://www.bu.edu/peaclab/publications` after the camera-ready deadline.

2 Included Files

- `requirements.txt`: A list of Python packages required.
- `README.pdf`: This document.
- `notebook.ipynb`: The interactive Jupyter Notebook file. Its operation is outlined in this document.
- `notebook.html`: A static HTML version of the notebook that can be viewed by a browser.
- `taxonomist.py`: A Python file including a basic version of the Taxonomist framework. The module contents can be imported for other projects.
- `data/`: The monitoring data collected from different applications executed on Volta.
  - `metadata.csv`: A csv file listing each run, the IDs of the nodes where each run executed on, which application was executed with which inputs, the start and end times and the duration of the applications.

- `timeseries.tar.bz2`: A bzip2 compressed file containing the data collected. The uncompressed size is 16 GB, it is not necessary to uncompress for most of the notebook.
- `features.hdf`: A HDF5 File containing the pre-calculated features. The calculation process is included in the notebook.

3 Installation Steps

- After downloading the zip file, extract it using `unzip artifact.zip`.
- We have tested this notebook on Linux (Fedora 27 and Ubuntu). Although the Python code and packages should work on Windows and Mac OS, we have not performed tests on these platforms.
- Python is required to run this notebook. We have tested it with Python 3.6.5; however, older versions of Python 3 should also work.
- We recommend using a Python package manager to install the packages. We have used pip with Python virtualenv to manage the packages.
- If pip is used, the required packages can be installed by `pip install -r requirements.txt`. If no virtualenv is used, the `--user` flag is recommended for pip.
- If a desktop computer is used, Jupyter will launch a web browser that can be used to view the notebook. The instructions for running Jupyter on a remote server and accessing it using a local browser can be found at `http://Jupyter-notebook.readthedocs.io/en/stable/public_server.html`.

4 Getting Started

After installing the required Python packages, start the Jupyter process (e.g., by executing `Jupyter notebook`). After opening `notebook.ipynb` using a browser, the individual cells can be executed. There are some global variables in the notebook that specify the code behavior, the descriptions of these variables are given in the comments in the code in the notebook.

5 Differences from the Paper

For clarity purposes and due to lack of space, there are a few omissions in the notebook compared to the code used for our Euro-Par paper. These differences are:

- The dataset included with the notebook only includes one third of the runs with benchmark applications. However, all of the input sizes are included.

- To keep the running time within the required duration of several hours, the hyperparameter tuning is not performed (Sec. 4.3).
- In the paper, we use five-fold cross validation for evaluation. In this notebook, only the first fold is used.
- The data from bitcoin miners or applications that ran on Volta by other users are not included.
- The tests with unknown inputs or classifiers other than random forest are not included; however, such tests can be performed with minimal modification to the code in the notebook.
- Setting up the package dependencies for the baseline method (Sec. 5.3.) is not straightforward; hence, the baseline is not included.
- The code for deriving and plotting the importance of individual features and metrics (Sec. 6.) is not included.

Acknowledgment

This work has been partially funded by Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energys National Nuclear Security Administration under Contract DENA0003525.